CVXR: An R Package for Disciplined Convex Optimization

Anqi Fu  Bala Narasimhan  Stephen Boyd
EE & Statistics Departments
Stanford University

useR! Conference 2016
Outline

Convex Optimization

CVXR

Examples

Future Work
Convex Optimization

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, M \\
& \quad Ax = b
\end{align*}
\]

with variable \( x \in \mathbb{R}^n \)

- Objective and inequality constraints \( f_0, \ldots, f_M \) are convex
- Equality constraints are linear
Convex Optimization

minimize \( f_0(x) \)
subject to \( f_i(x) \leq 0, \quad i = 1, \ldots, M \)
\( Ax = b \)

with variable \( x \in \mathbb{R}^n \)

- Objective and inequality constraints \( f_0, \ldots, f_M \) are convex
- Equality constraints are linear

Why?
- We can solve convex optimization problems
- There are many applications in many fields, including machine learning and statistics
Convex Problems in Statistics

- Least squares, nonnegative least squares
- Ridge and lasso regression
- Isotonic regression
- Huber (robust) regression
- Logistic regression
- Support vector machine
- Sparse inverse covariance
- Maximum entropy and related problems
- ... and new methods being invented every year!
Domain Specific Languages for Convex Optimization

- Special languages/packages for general convex optimization
- CVX, CVXPY, YALMIP, Convex.jl
- Slower than custom code, but extremely flexible and enables fast prototyping

```python
from cvxpy import *

beta = Variable(n)
cost = norm(X * beta - y)
prob = Problem(Minimize(cost))
prob.solve()
beta.value
```
Domain Specific Languages for Convex Optimization

- Special languages/packages for general convex optimization
- CVX, CVXPY, YALMIP, Convex.jl
- Slower than custom code, but extremely flexible and enables fast prototyping

```python
from cvxpy import *
beta = Variable(n)
cost = norm(X * beta - y)
prob = Problem(Minimize(cost))
prob.solve()
beta.value
```
Outline

Convex Optimization

CVXR

Examples

Future Work
CVXR

A modeling language in R for convex optimization

- Connects to many open source solvers
- Uses disciplined convex programming to verify convexity
- Mixes easily with general R code and other libraries
Outline

Convex Optimization

CVXR

Examples

Future Work
Ordinary Least Squares (OLS)

- minimize $||X\beta - y||_2^2$
- $\beta \in \mathbb{R}^n$ is variable, $X \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$ are constants
Ordinary Least Squares (OLS)

- minimize $||X\beta - y||^2_2$
- $\beta \in \mathbb{R}^n$ is variable, $X \in \mathbb{R}^{m\times n}$ and $y \in \mathbb{R}^m$ are constants

library(CVXR)
beta <- Variable(n)
obj <- sum_squares(y - X %*% beta)
prob <- Problem(Minimize(obj))
result <- solve(prob)
result$value
result$getValue(beta)

- $X$ and $y$ are constants; beta, obj, and prob are S4 objects
- solve method returns a list that includes optimal beta and objective value
Non-Negative Least Squares (NNLS)

- minimize $||X\beta - y||_2^2$ subject to $\beta \geq 0$
Non-Negative Least Squares (NNLS)

- minimize \( |X\beta - y|^2 \) subject to \( \beta \geq 0 \)

\[
\begin{align*}
\text{constr} &\leftarrow \text{list(beta} \geq 0) \\
\text{prob2} &\leftarrow \text{Problem(Minimize(obj), constr)} \\
\text{result2} &\leftarrow \text{solve(prob2)} \\
\text{result2$value} &
\text{result2$getValue(beta)}
\end{align*}
\]

- Construct new problem with list \( \text{constr} \) of constraints formed from constants and variables

- Variables, parameters, expressions, and constraints exist outside of any problem
## True vs. Estimated Coefficients

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>True Estimate</th>
<th>NNLS Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>b0</td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>b1</td>
<td>0.0</td>
<td>1.5</td>
</tr>
<tr>
<td>b2</td>
<td>0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>b3</td>
<td>1.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>b4</td>
<td>1.5</td>
<td>-1.5</td>
</tr>
</tbody>
</table>

**Diagram:**

The diagram illustrates the comparison between true and estimated coefficients for various types of estimation methods (NNLS, True, OLS). The x-axis represents the coefficient (b0 to b9), and the y-axis shows the estimate ranging from -1.5 to 1.5. Each coefficient is marked with bars indicating the true and estimated values, allowing for a visual comparison of accuracy across different estimation techniques.
True vs. Estimated Coefficients

Examples
Direct Standardization

- Samples \((X, y)\) drawn **non-uniformly** from a distribution
- Expectations of columns of \(X\) have known values \(b \in \mathbb{R}^n\)
Direct Standardization

- Samples \((X, y)\) drawn **non-uniformly** from a distribution
- Expectations of columns of \(X\) have known values \(b \in \mathbb{R}^n\)
- Empirical distribution \(y = y_i\) w.p. \(1/m\) is **not** a good estimate of distribution of \(y\)
- Let’s use weighted empirical distribution \(y = y_i\) w.p. \(w_i\)
- Choose \(w = (w_1, \ldots, w_m)\) to match known expectations, maximize entropy

\[
\text{maximize } \sum_{i}^{m} -w_i \log w_i \\
\text{subject to } w \geq 0, \quad \mathbf{1}^T w = 1, \quad X^T w = b
\]
Direct Standardization

\[ w \leftarrow \text{Variable}(m) \]
\[ \text{obj} \leftarrow \text{sum(entr}(w)) \]
\[ \text{constr} \leftarrow \text{list}(w \geq 0, \text{sum}(w) = 1, \text{t}(X) \text{%*% w} = b) \]
\[ \text{prob} \leftarrow \text{Problem(Maximize(obj), constr)} \]
\[ \text{result} \leftarrow \text{solve(prob)} \]
\[ \text{result}\$\text{getValue}(w) \]

- \text{entr} is the elementwise entropy function
- \text{result}\$\text{getValue}(w) returns an R vector of weights
True vs. Estimated Cumulative Distribution

Examples
True vs. Estimated Cumulative Distribution

Examples
Outline

Convex Optimization

CVXR

Examples

Future Work
Future Work

- Connect to more solvers: MOSEK, GUROBI, …
- Flesh out convex functions in library
- Develop more applications and examples
- Add warm start support

Github repo: https://github.com/anqif/cvxr